

## Converting Plane Stress Statics to 2D Natural Frequencies, changes in blue

```

function Plane_Stress_T6_Frequency (n_modes)
%.....
% Plane Stress Frequency Analysis (2.5D), Isoparametric T6
% (deleted reference to load vectors)
%.....
if ( nargin == 0 )           ; % check for optional data
    n_modes = 3             ; % modes to be output
end % if from argument count
echo_user_remarks % optional use of new file msh_remarks.tmp

%           Application and element dependent controls
n_g = 2 ; % number of DOF per node
n_r = 3 ; % number of rows in B_e matrix

%           Read mesh input data files
[n_m, n_s, P, x, y, z] = get_mesh_nodes ;

%           Extract EBC flags from packed integer flag P
[EBC_flag] = get_ebc_flags (n_g, n_m, P) ; % unpack flags
EBC_count = sum( sum ( EBC_flag > 0 ) ) ; % # of EBC

[n_e, n_n, n_t, el_type, nodes] = get_mesh_elements ;
n_d = n_g*n_m ; % system degrees of freedom (DOF)
n_i = n_g*n_n ; % number of DOF per element
S = zeros (n_d, n_d) ; C = zeros (n_d, 1) ; % initialize sums
M = zeros (n_d, n_d) ; ; % initialize sums

%           Read EBC values, if any
if ( EBC_count > 0 ) ; % need EBC data
    [EBC_value] = get_ebc_values (n_g, n_m, EBC_flag) ; % read data
end % if any EBC data expected

%           Load the element properties array
load msh_properties.tmp ; % one row per element
n_prop = size(msh_properties, 1) ; % == 1 if homogeneous
fprintf ('\n(Echoing columns of file msh_properties.tmp) \n')
if ( n_prop == 1 )
    E = msh_properties (1, 1) ; % modulus of elasticity
    nu = msh_properties (1, 2) ; % Poisson's ratio
    Body_x = msh_properties (1, 3) ; % x-body force component
    Body_y = msh_properties (1, 4) ; % y-body force component
    Thick = msh_properties (1, 5) ; % thickness
    Rho = msh_properties (1, 6) ; % mass density

%           ECHO PROPERTIES
    fprintf ('\nProperties for all elements \n')
    fprintf ('Elastic Modulus = %g \n', E)
    fprintf ('Poisson's ratio = %g \n', nu)
    fprintf ('X-body force component = %g \n', Body_x)
    fprintf ('Y-body force component = %g \n', Body_y)
    fprintf ('Thickness = %g \n', Thick)
    fprintf ('Mass Density = %g \n', Rho)
end % if
if ( n_prop > n_e )
    error ('ERROR: number of property sets exceeds number of elements')
end % if

```

## Converting Plane Stress Statics to 2D Natural Frequencies, changes in blue

```

%   Tabulate quadrature data for unit triangle
n_q = 7 ; % exact for 4-th degree polynomials
r_q = [ 0.10128651, 0.47014206, 0.79742699, 0.47014206, ...
        0.10128651, 0.05971587, 0.33333333 ] ;
s_q = [ 0.10128651, 0.05971587, 0.10128651, 0.47014206, ...
        0.79742699, 0.47014206, 0.33333333 ] ;
w_q = [ 0.06296959, 0.06619708, 0.06296959, 0.06619708, ...
        0.06296959, 0.06619708, 0.11250000 ] ;

%   GENERATE ELEMENT MATRICES AND ASSYMBLE INTO SYSTEM
%   Assemble n_d by n_d square matrix terms from n_e by n_e

for j = 1:n_e ; % element loop ==> ==> ==> ==> ==> ==>
    e_nodes = nodes ( j, 1:n_n) ; % connectivity

%   SET ELEMENT PROPERTIES & GEOMETRY
if ( n_prop > 1 )
    E = msh_properties ( j, 1) ; % modulus of elasticity
    nu = msh_properties ( j, 2) ; % Poisson's ratio
    Body_x = msh_properties ( j, 3) ; % x-body force component
    Body_y = msh_properties ( j, 4) ; % y-body force component
    Thick = msh_properties ( j, 5) ; % thickness
    Rho = msh_properties ( j, 6) ; % mass density

%   ECHO PROPERTIES
fprintf ('\nProperties for all elements \n')
fprintf ('Elastic Modulus = %g \n', E)
fprintf ('Poisson's ratio = %g \n', nu)
fprintf ('X-body force component = %g \n', Body_x)
fprintf ('Y-body force component = %g \n', Body_y)
fprintf ('Thickness = %g \n', Thick)
fprintf ('Mass Density = %g \n', Rho)
if ( Thick <= 0 )
    error ('\nError: zero thickness at element %g ', j)
end % if invalid data
end % if
E_e = E / (1 - nu^2) * [ 1, nu, 0 ; ...
                      nu, 1, 0 ; ...
                      0, 0, (1 - nu)/2 ] ; % stress-strain relation

%   Gather nodal coordinates
xy_e (1:n_n, 1) = x(e_nodes) ; % x coord at el nodes
xy_e (1:n_n, 2) = y(e_nodes) ; % 6 x 2 % y coord at el nodes

%   Numerical Intergation of S_e, C_e
S_e = zeros (n_i, n_i) ; % initalize values
B_q = zeros (n_r, n_i) ; % initalize values
M_e = zeros (n_i, n_i) ; % initalize values
M_x = zeros (n_n, n_n) ; % initalize values

for q = 1:n_q ; % begin loop ----> ----> ----> ----> ----> ----> ---->
    r = r_q (q) ; s = s_q (q) ; w = w_q (q) ; % recover data

```

## Converting Plane Stress Statics to 2D Natural Frequencies, changes in blue

```

%           element type interpolation functions, 1 x 6
H_q = [ (1 - 3*r - 3*s + 2*r*r + 4*r*s + 2*s*s), (2*r*r - r), ...
        (2*s*s - s), 4*(r - r*r - r*s), 4*r*s, 4*(s - r*s - s*s) ] ;
DLH_q = [ (-3 + 4 * r + 4 * s), (-1 + 4 * r), 0, ...
          (4 - 8 * r - 4 * s), (4 * s), (-4 * s) ; ... % dH/dr
          (-3 + 4 * r + 4 * s), 0, (-1 + 4 * s), ...
          (-4 * r), (4 * r), (4 - 4 * r - 8 * s) ] ; % dH/ds

%           Global position and Jacobian
xy_q      = H_q * xy_e           ; % 1 x 2      % global (x, y)
Jacobian  = DLH_q * xy_e        ; % 2 x 2      % d(x,y)/d(r,s)
J_det     = abs( det (Jacobian) ) ; % 1 x 1      % determinant
J_Inv     = inv (Jacobian)      ; % 2 x 2      % inverse

%           Global derivatives of H
DGH_q = J_Inv * DLH_q          ; % 2 x 6      % dH/dx & dH/dy
E_q = E_e                      ; % 3 x 3      % properties
t_q = Thick                    ; % 1 x 1      % thickness

%           Insert gradient terms into the differential operator
B_q (1, 1:2:11) = DGH_q (1, 1:6) ; % for esplion_x
B_q (2, 2:2:12) = DGH_q (2, 1:6) ; % for esplion_y
B_q (3, 1:2:11) = DGH_q (2, 1:6) ; % for gamma
B_q (3, 2:2:12) = DGH_q (1, 1:6) ; % for gamma

%           ELEMENT MATRICES UPDATES, 12 x 12 and 12 x 1
S_e = S_e + (B_q' * E_q * B_q) * t_q * J_det * w_q (q) ; % square
M_x = M_x + (H_q' * Rho * H_q) * t_q * J_det * w_q (q) ; % mass
end % for quadratures <---- <---- <---- <---- <---- <---- <---- <----
M_e (1:2:11, 1:2:11) = M_x ; % repeat mass for x-motion
M_e (2:2:12, 2:2:12) = M_x ; % repeat mass for y-motion

%           SCATTER TO (ASSEMBLE INTO) SYSTEM ARRAYS
% Insert completed element matrices into system matrices
[rows] = get_element_index (n_g, n_n, e_nodes); % eq numbers
S (rows, rows) = S (rows, rows) + S_e ; % add to system sq
M (rows, rows) = M (rows, rows) + M_e ; % add to system sq
end % for elements <<===== <<===== <<===== <<===== <<===== <<===== <<=====

%           ENFORCE ESSENTIAL BOUNDARY CONDITIONS
if ( EBC_count > 0 ) % then no rigid body motion modes
  Ignore = -987654 ; % flag for setting eigenvalues to ignore
  [S, M] = eigen_essential_BC (EBC_flag, Ignore, S, M) ; % apply EBC
  % above 2 lines not needed if all unknowns numbered first
end % if

%           COMPUTE SOLUTION & SAVE
[V,D] = eig(S, M) ; % solve full general eigenproblem

% If all the unknown dof are numbered first, its more efficient to use
% n_unkn = n_d - EBC_count ; % ASSUME all unknowns numbered first
% if ( n_unkn < n_modes )
%   n_modes = n_unkn
% end % if
% [V,D] = eig(S(1:n_unkn, 1:n_unkn), M(1:n_unkn, 1:n_unkn)) ; % faster

```

## Converting Plane Stress Statics to 2D Natural Frequencies, changes in blue

```

V = real(V) ; D = real(D)           ; % clean up. Should be real
for j = 1:n_d                       ; % extract diagonal matrix
    F_sq (j) = D (j,j)              ; % (fake bc values = Ignore)
end % for j
format long
[F_sq, Order] = sort (F_sq)         ; % reorder eigenvalues, lo to hi

% list the results
kount = 0                           ; % number of true eigenvalues
for j = 1:n_d                       ; % list eigenvalues only
    if ( F_sq(j) ~= Ignore )        ; % then not essential BC dof
        kount = kount + 1          ; % number of true eigenvalues
        Freq = sqrt(F_sq(j))/(2*pi); % frequency, cycles per second
        fprintf ('Natural frequency (Hz) %i = %g \n', ...
                kount, Freq )      ; % values
    end % if a true frequency
    if ( kount == n_modes ) ;
        break % out of this for loop
    end % if
end % for j

show = n_modes                      ; % option to show eigenvectors
if ( show > 0 )                    ; % open for writing
    fid = fopen('mode_value_vector.tmp', 'w') ;
    kount = 0                       ; % number of true eigenvalues
    for j = 1:n_d                   ; % list eigenvalues only
        if ( F_sq(j) ~= Ignore )    ; % then not essential BC dof
            kount = kount + 1        ; % number of true eigenvalues
            Freq = sqrt(F_sq(j))/(2*pi); ; % frequency
            actual = Order(j)        ; % where in the sorted list
            fprintf ('Frequency (Hz) %i = %g \n', kount, Freq); % values
            fprintf ('Eigenvector (mode shape): \n')           ; % vectors
            fprintf ('x-component, y_component \n')           ; % vectors
            results = reshape ( V(:, actual), n_g, n_m)'       ; % x-y form
            biggest = max ( max ( abs (results) ) )             ; % scale
            results = results / biggest                         ; % scaled
            disp (results)                                     ;
        end
    end

% save results for mode shape plots
    for k = 1:n_m                                     ; % loop over all nodes
        fprintf (fid, '%i %g %g %g \n', ...
                kount, Freq, results (k, 1), results (k, 2))
    end % for result save

    if ( kount == n_modes ) ; % break out of this for loop
        break ;
    end % if finished requested number of modes
end % if a true mode, not associated w a EBC dof
end % for j over all degrees of freedom
fclose(fid) ; % done with all requested plot data saves
end % if list

% End finite element calculations.
% See /home/mech517/public_html/Matlab_Plots for graphic options
% http://www.owl.net.rice.edu/~mech517/help_plot.html for help
% end of function Plane_Stress_T6_Frequency

```

## Converting Plane Stress Statics to 2D Natural Frequencies, changes in blue

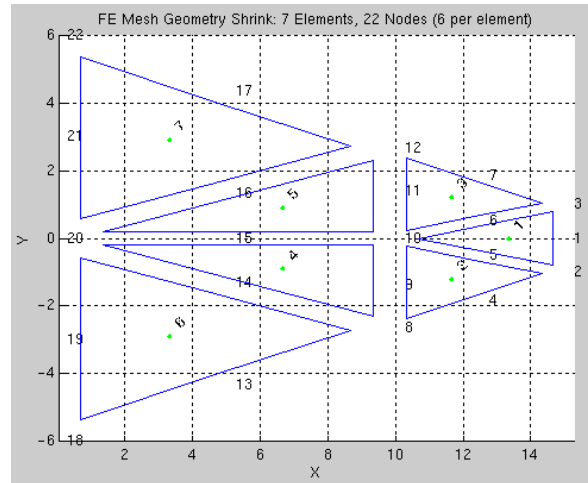
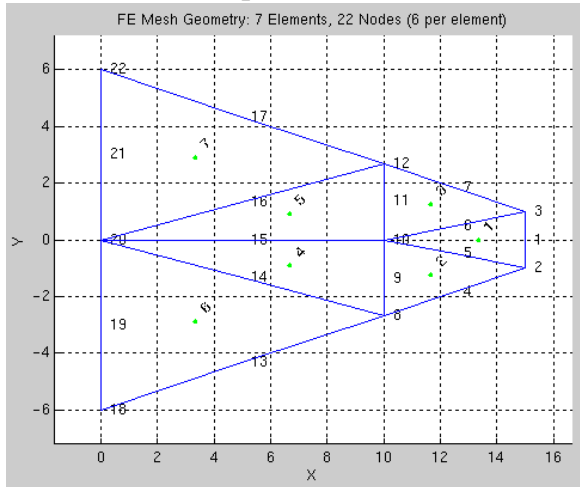
### Validation run

```
>> addpath /net/course-a/mech517/public_html/Matlab_Plots
```

```
>> mesh_plot
```

```
Read 22 mesh coordinate pairs. Read 7 elements with 6 nodes each
```

```
>> mesh_shrink_plot
```



```
>> Plane_Stress_T6_frequency(4)
```

(Echo of msh\_remarks.tmp)

Natural frequency of a tapered beam.

Material brass.

Fixed - pinned ends

Compare to "Formulas for Natural Frequency & Mode Shape"

by Blevins, Fig. 8-19

Read 22 nodes.

(Echo of file msh\_bc\_xyz.tmp)

bc\_flag, x-, y-coordinates

```
11 15 0
00 15 -1
00 15 1
00 12.5 -1.833
00 12.5 -0.5
00 12.5 0.5
00 12.5 1.833
00 10 -2.667
00 10 -1.375
00 10 0
00 10 1.375
00 10 2.667
00 5 -4.333
00 5 -1.333
00 5 0
00 5 1.333
00 5 4.333
11 0 -6
11 0 -3
11 0 0
11 0 3
```

## Converting Plane Stress Statics to 2D Natural Frequencies, changes in blue

11 0 6

(Echo of file msh\_typ\_nodes.tmp)

Read 7 elements with (ignored) type & 6 nodes each.

1	2	3	10	1	6	5
1	8	2	10	4	5	9
1	10	3	12	6	7	11
1	8	10	20	9	15	14
1	10	12	20	11	16	15
1	18	8	20	13	14	19
1	20	12	22	16	17	21

Applied Displacement Boundary Conditions: 12

(Echo of file load msh\_ebc.tmp)

Node, DOF (1=u, 2=v, 3=r), Value.

1	1	0
1	2	0
18	1	0
19	1	0
20	1	0
21	1	0
22	1	0
18	2	0
19	2	0
20	2	0
21	2	0
22	2	0

(Echoing columns of file msh\_properties.tmp)

Properties for all elements

Elastic Modulus	=	1e+11
Poisson's ratio	=	0.33
X-body force component	=	0
Y-body force component	=	0
Thickness	=	1
Mass Density	=	8500

Natural frequency (Hz) 1 = 52.3112

Natural frequency (Hz) 2 = 107.995

Natural frequency (Hz) 3 = 112.327

Natural frequency (Hz) 4 = 208.705

Frequency (Hz) 1 = 52.3112

Eigenvector (mode shape):

x\_component, y\_component

	0	0
-0.29497729608831		0.00787598556598
0.29497729519989		0.00787598514531
-0.22998794688469		0.77743786213498
-0.06907255521933		0.79523176937558
0.06907255398872		0.79523176950069
0.22998794608075		0.77743786207141
0.04116517315606		0.98931016710651
-0.01708444962725		0.99999999978127
-0.00000000017687		0.99989853632782
0.01708444776027		1.00000000000000

Converting Plane Stress Statics to 2D Natural Frequencies, changes in blue

-0.04116517527959	0.98931016745942
0.19026482769648	0.53032163544070
0.03593409909354	0.54494320543470
-0.00000000017399	0.55314928499829
-0.03593409996644	0.54494320534016
-0.19026482899090	0.53032163569119
0	0
0	0
0	0
0	0
0	0

Frequency (Hz) 2 = 107.995

Eigenvector (mode shape):

x-component, y\_component

0	0
0.26520282289455	-0.04611953258454
0.26520281601056	0.04611953255637
0.82950459703056	-0.09823196179123
0.71693224285308	-0.04024822231382
0.71693224441931	0.04024820326112
0.82950459689238	0.09823194375443
0.99744403755503	-0.05996215921777
0.99999999506775	-0.04043145660446
0.98803839027058	-0.00000000460062
1.00000000000000	0.04043144693531
0.99744404974790	0.05996214985148
0.54427938299956	0.13747750901604
0.72409162947443	0.04475612745107
0.76377960625529	0.00000000744726
0.72409163069098	-0.04475611279049
0.54427938942194	-0.13747749590277
0	0
0	0
0	0
0	0
0	0
0	0

Frequency (Hz) 3 = 112.327

Eigenvector (mode shape):

x-component, y\_component

0	0
0.37457045128507	-0.02452446792161
-0.37457045462373	-0.02452446765080
-0.00118233281321	-0.95957827555328
0.02106613941823	-0.99999999903800
-0.02106614922787	-1.00000000000000
0.00118231988647	-0.95957827710579
-0.62636881047423	-0.50406573327587
-0.25181685130372	-0.54171275715869
-0.00000000797261	-0.55604017613948
0.25181683713304	-0.54171275876215
0.62636879818848	-0.50406573570238
-0.27641912345390	0.59025861062446
-0.13802345262758	0.65880573249900
-0.00000000681424	0.65997344811191
0.13802344033501	0.65880573314446

## Converting Plane Stress Statics to 2D Natural Frequencies, changes in blue

```

0.27641911650160    0.59025861083543
                    0                0
                    0                0
                    0                0
                    0                0
                    0                0
                    0                0

```

Frequency (Hz) 4 = 208.705

Eigenvector (mode shape):

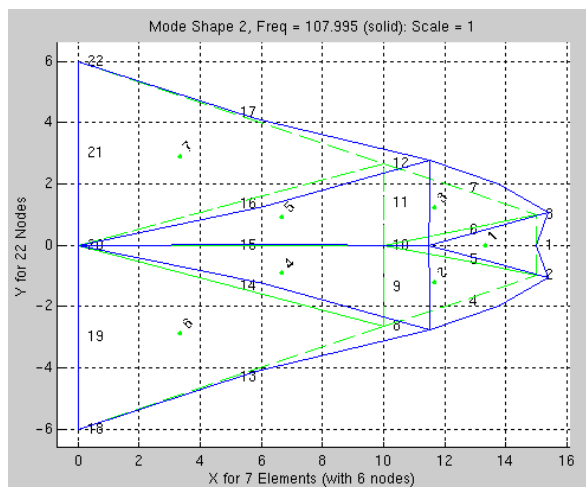
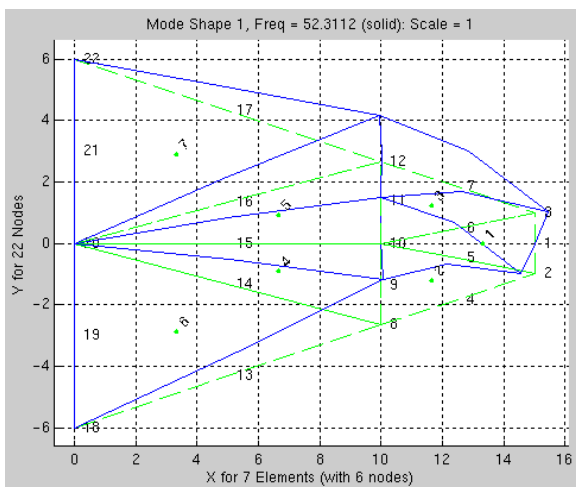
```

x-component,  y_component
                    0                0
-0.24536814735684 -0.00804728782143
 0.24536815751015 -0.00804728529275
-0.58988394769963  0.27448854386331
-0.19910990520204  0.25810878939915
 0.19910992617783  0.25810879042709
 0.58988396383152  0.27448854400688
-0.99999998159637  0.31605744642658
-0.64669917191066  0.33612138402351
 0.00000001345827  0.34499913583645
 0.64669918403228  0.33612138068617
 1.00000000000000  0.31605744002585
-0.99206988294904 -0.30055116425256
-0.59382039468131 -0.11945385702581
-0.00000000153873 -0.11208123422262
 0.59382038774371 -0.11945385715791
 0.99206987265114 -0.30055116123003
                    0                0
                    0                0
                    0                0
                    0                0
                    0                0

```

>> mode\_2d\_mesh\_plot(1,1,1,1)

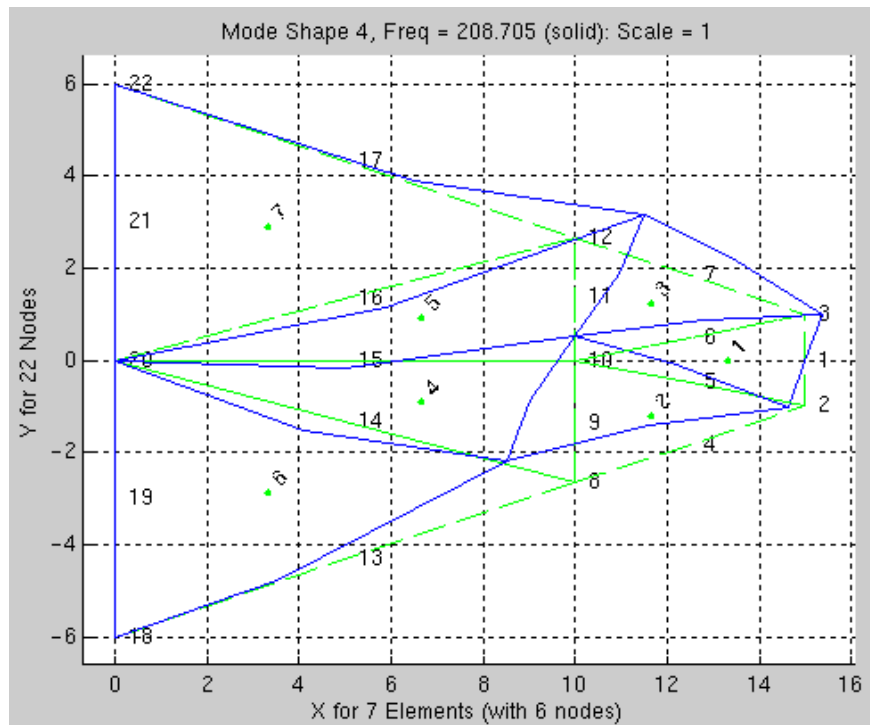
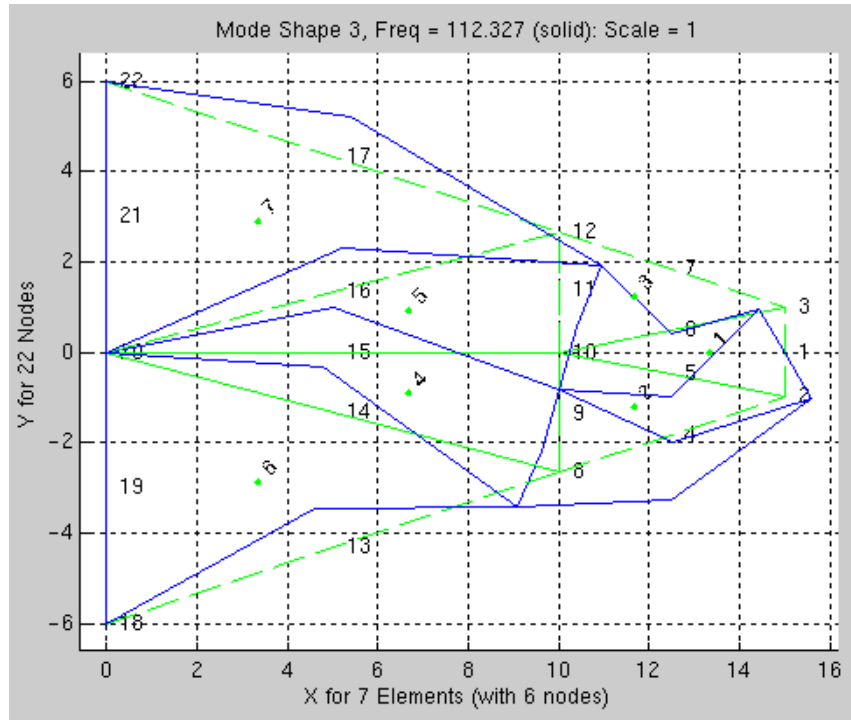
>> mode\_2d\_mesh\_plot(2,1,1,1)



Converting Plane Stress Statics to 2D Natural Frequencies, changes in blue

```
>> mode_2d_mesh_plot(3,1,1,1)
```

```
>> mode_2d_mesh_plot(4,1,1,1)
```



```
>> quit
```